Fișe de lucru la *Sisteme de operare pentru servere* (M3)

clasa a XII-a
Profilul tehnic
Domeniul Electronică-automatizări
Calificarea Tehnician operator tehnică de calcul

Prof. ing. Popescu Călin
Liceul Tehnologic „Ferdinand I" Curtea de Argeș

Cuprins

Exemple de comenzi pentru Linux Debian/Ubuntu ……5

Configurarea unui server DHCP cu Ubuntu Linux……..8

Configurarea unui server DNS cu Ubuntu Linux…….. 10

Partajarea de directoare cu Samba…….13

Găzduirea unui site cu serverul Apache……15

Utilizarea programului utilitar pentru firewall al Ubuntu Linux, *ufw*……17

Rularea containerelor cu *Docker*, în Linux Ubuntu……..19

Virtual Host cu Apache (două adrese de site pe aceeași adresă IP)……22

Instalarea serverului de baze de date MariaDB și lucrul cu bazele de date……23

Instalarea Joomla în Linux Ubuntu, alături de Apache și MariaDB……...26

Cum ne conectăm prin ssh cu cheie la o instanță EC2, în AWS…… 29

Crearea unui domeniu pentru o rețea locală cu Windows Server……..30

Instalarea de software dintr-un repository cu utilitarul apt:

sudo apt update – actualizează indexul programelor din depozitele externe (repositories), pentru a asigura că instalăm cele mai noi versiuni pentru software-ul dorit

apt search gimp – caută în repositories aplicații cu nume apropiat de gimp

apt show gimp – afișează mai multe informații despre aplicația cu numele gimp (pentru a vedea dacă e cea pe care o căutam)

sudo apt install gimp – instalează din repository aplicația **gimp**

sudo apt remove gimp – dezinstalează aplicația **gimp**

Instalarea de software din pachete .deb descărcate, cu utilitarul *dpkg*:

sudo dpkg -i – instalează un pachet de tip .deb, de ex:

sudo dpkg -i google-chrome-stable_current_amd64.deb

sudo dpkg -l – listează toate pachetele instalate în prezent pe sistemul propriu – chiar și cele instalate cu **apt.**

sudo dpkg -r – șterge un pachet; are nevoie însă ca argument de numele exact folosit de linux pentru pachetul respectiv; acesta se poate găsi, de pildă cu:

dpkg -l | grep google – filtrând cu **grep** rezultatele ce conțin google, cu vom găsi: google-chrome-stable; vom folosi deci:

sudo dpkg -r google-chrome-stable – dezinstalează aplicația google chrome, pe care o instalasem dintr-un pachet .deb

dpkg -L apache2 – arată toate fișierele care aparțin pachetului **apache2**

Montarea unui sistem extern de fișiere (stick USB, CD etc)

sudo mount /dev/sdc1 /mnt/ptdisc – montează dispozitivul sdc1 în directorul ptdisc din /mnt (numele exact sub care apare dispozitivul nou-conectat în directorul /dev se identifică fie cu consola Cockpit, fie folosind comanda **lsblk**)

sudo umount /mnt/ptdisc sau **sudo umount /dev/sdc1** – de-montează dispozitovul sdc1

Mutare, copiere, ștergere, creare de fișiere și directoare

mv abc def – redenumește fișierul abc ca def, în același director

mv ghe ~ – mută fișierul ghe în propriul director din /home (notat și ~)

mv /home/ferdi3/dirmeu/ /home/ferdi3/Documents/ – mută directorul dirmeu, cu tot cu conținutul lui, în directorul home/ferdi3/Documents

cp fis1 fis1d – copiază fișierul fis1 în fișierul fis1d, în același director

cp -r /home/ferdi2/dirmeu /tmp/a/ – copiază directorul dirmeu în /tmp/a/, cu tot conținutul

rm fis1 – șterge fișierul fis1

rm * – șterge toate fișirele din directorul curent

rmdir /home/ferdi4/dirmeu/ – șterge directorul dirmeu, dacă este gol

rm -r /home/ferdi4/dirmeu/ – șterge directorul dirmeu, cu tot cu conținut

mkdir /home/ferdi9/undir – creează un director undir în /home/ferdi9

touch fisiergol – creează fișierul gol fisiergol (în directorul curent)

nano fisnou – deschide (în directorul curent) cu editorul nano fișierul fisnou (îl creează, dacă nu exista)

Arhivare și comprimare fișiere

tar -cf arhiva.tar fis1 fis2 fis3 --remove-files – creează o arhivă în fișierul cu numele arhiva.tar, din trei fișiere (pe care apoi le și șterge)

tar -czf arhiva.tar.gz fis4 fis5 fis6 – creează arhivă, dar și comprimă cu gzip

tar -cjf arhiva.tbz fis4 fis5 fis6 – creează arhivă, dar comprimă cu bzip2

tar -t arhiva.tar **–** listează conținutul din arhiva.tar

tar -j arhiva.tbz – decomprimă arhiva, cu cu bzip2

Alte operații cu fișiere

cd / cd .. cd /var/www/html –

cat – citește un fișier; **head –** citește primele 10 linii, **tail -5 –** citește ultimele (5) linii

echo scrie ceva pe ecran – scrie pe ecran

echo scrie ceva într-un fișier > fisasta – scrie într-un fișier (cu >>, scrie pe rândul următor)

ls -lh – listează conținutul directorului (varianta lungă, human-readable)

ls -lS – listează în varianta lungă și sortează

chmod u+x fis – face fișierul fis executabil pentru userul deținător; alte ex de opțiuni: **ug+w** (*userul* și *grupul* care dețin fișierul primesc dreptul de *write*), **o-r** (*others* sunt lipsiți de dreptul de read), **a=rx** (*all* au permisiunile *read* și *execute*)

chown ferdi /cale/fisier sau **chown ferdi:grupferdi /cale/fisier** sau **chown :grupferdi /cale/fisier –** face deținător al fișierului fisier un user, un user și un grup, sau un grup

Adăugarea și ștergerea de useri

useradd -d /home/calin -m calin – creează un user calin care are și director în home, chiar deodată ce e creat

passwd – se adaugă parolă userului

useradd calin – creează doar userul

ștergerea se poate face întâi cu sudo **userdel calin** și apoi i se poate șterge și directorul din home cu **rm -r /home/calin**

sau se poate șterge direct totul, cu: **sudo -r userdel calin**

adduser calin – creează un user **calin**, la fel, dar cu un dialog la promptul de comandă)

se poate șterge și cu **deluser --remove-home calin**

sudo usermod -aG sudo calin – face userul calin membru al grupului sudo

su calin – userul curent se schimbă cu userul calin (**su** simplu schimbă cu userul root)

sudo calin – execută o comandă cu drepturile userului calin (**sudo** simplu – cu drepturile de root)

Administrarea calculatorului cu consola Cockpit

sudo apt install cockpit – instalează cockpit; se activează cu:

systemctl enable --now cockpit.socket ; apoi în se deschide în browser: **localhost:9090**

Accesul la distanță cu Secure Shell (*ssh*)

sudo install openssh-server – instalează serverul ssh

ssh ferdi2@192.168.10.15 – acces la linia de comandă a 192.168.10.15, cu contul ferdi2

scp -r ferdi3@192.168.10.14:/usr/share/man/man1/ /tmp – copiere de director de la server la calculatorul propriu

scp home/ferdi4/Documents/fis1 ferdi3@192.168.10.14:/home/ferdi3 – copiere de fișier de la calculatorul propriu la server, în directorul userului ferdi3

Pentru a muta traficul ssh de pe portul implicit 22 pe alt port (de ex 65535) deschidem fișierul /etc/ssh/sshd_config și sub linia comentată care indică Port 22, adăugăm: Port 65535

După ce salvăm repornim ssh: sudo systemctl restart ssh

Apoi, vom accesa serverul indicând portul:

ssh -p 65535 ferdi2@192.168.10.15

pentru copiere de directoare sau fișiere, portul e indicat cu majusculă:

sau scp -P 65535 -r ferdi3@192.168.10.14:/usr/share/man/man1/ /tmp

Pentru configurări de rețea fără interfața grafică, accesăm de la linia de comandă utilitarul: **nmtui**

Instalăm pachetul ***isc-dhcp-server***:
sudo apt install isc-dhcp-server
îi verificăm starea cu
systemctl status isc-dhcp-server
(se afișează un mesaj, din care ieșim cu **Ctrl+C**)

Pentru a opri temporar serviciul:
sudo systemctl stop isc-dhcp-server
pentru a-l dezactiva:
sudo systemctl disable isc-dhcp-server
pentru a opri și dezactiva serviciul dhcp pentru adrese IPv6:
sudo systemctl status isc-dhcp-server6
sudo systemctl disable isc-dhcp-server6

Pentru a păstra fișierul configurat din oficiu, cu alt nume, îl redenumim:
sudo mv /etc/dhcp/dhcpd.conf /etc/dhcp/dhcpd.conf.orig

Creăm apoi un fișier pe care îl vom pune în locul fișierul ***dhcpd.conf*** din oficiu, în același director /***etc/dhcpd***. În interiorul fișierului vom avea un conținut după acest model:

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
authoritative;
subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.120 192.168.1.140;
option routers 192.168.1.1;
option domain-name-servers 8.8.8.8;
}
```

(în cazul de mai sus, se alocă adrese IP între 192.168.1.120 și 192.168.1.140 - timpii fiind indicați în secunde)

Editām apoi și fișierul ***/etc/default/isc-dhcp-server***
unde completăm linia

INTERFACESv4=" "

Pentru a afla numele interfeței de rețea la care serverul va asculta cererile de adrese IP, dăm comanda **ip a** sau **ip addr show**, și reținem de acolo cum se cheamă interfața noastră de rețea, fie de wi-fi, fie de cablu. Și vom trece, după caz:

INTERFACESv4="eno1" sau **INTERFACESv4="enp0s3"** (în funcție de numele interfeței din calculatorul nostru).

După ce salvăm fișierul ***/etc/default/isc-dhcp-server*** cu linia respectivă modificată, repornim serverul nostru DHCP:

sudo systemctl start isc-dhcp-server

Și verificăm starea cu

systemctl status isc-dhcp-server

Dacă totul a fost setat corect (și fără greșeli de sintaxă), nu vom vedea mesaje de eroare, iar serverul dhcp va fi funcțional, în rețea.

Se instalează pachetul BIND (Berkeley Internet Name Daemon):
sudo apt install bind9

Setarea unui DNS extern

Pentru a reține în memoria cache răspunsurile unui server DNS extern, modificăm fișierul ***/etc/bind/named.conf.options***
(cu **sudo nano /etc/bind/named.conf.options)**
și schimbăm liniile următoare:

```
// forwarders {
// 0.0.0.0;
// };
```

astfel încât să arate astfel:

```
forwarders {
8.8.8.8;
8.8.4.4;
};
```

(partea verde poate lipsi; în locul acestor adrese pot fi și alte adrese ale unor servere DNS publice, sau ale furnizorului nostru de internet)

După ce salvăm, repornim serviciul ***bind9***:
sudo systemctl restart bind9
Și verificăm dacă totul e în regulă:
systemctl status bind9

Pentru a folosi noul nostru server DNS, trebuie să includem în fișierul de configurare al serverului dhcp o linie care să trimită la el.

De pildă, vom avea în ***/etc/dhcp/dhcpd.conf***, dacă adresa noastră este 10.1.1.8:

```
default-lease-time 28800;
max-lease-time 86400;
option subnet-mask 255.255.255.0;
option broadcast-address 10.1.1.255;
authoritative;
```

```
subnet 10.1.1.0 netmask 255.255.255.0 {
range 10.1.1.130 10.1.1.145;
option routers 10.1.1.1;
option domain-name-servers 10.1.1.8, 8.8.8.8;
}
```

Ceea ce putem observa este că timpul în care se identifică o adresă va scădea, a doua oară față de prima oară (căci serverul nostru a reținut-o deja în memoria sa cache).

Instalăm pachetul dns utils (**sudo apt install dnsutils**), dacă nu funcționează deja comanda **dig**. Apoi, scriem:

dig www.gov.ro de două ori, observând în răspuns linia de la sfârșit în care se declară timpul rezolvării cererii, de ex.:

;; Query time: 54 msec

A doua oară timpul va fi mai mic, de ex.:

;; Query time: 1 msec

Setarea unui DNS intern, și adăugarea de host-uri

Fișierul principal de configurare al ***bind*** se află la ***/etc/bind/named.conf***

El va avea la final, trei linii, prin care include și alte trei fișiere de configurare

```
include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

În ***named.conf.local***, trebuie să adăugăm la final un bloc de cod, pentru a indica unde se va găsi un fișier al zonei pe care o vom crea. Vom adăuga deci, de pildă, pentru o zonă numită ***oretea.buna***:

```
zone "oretea.buna" IN {
type master;
file "/etc/bind/net.oretea.buna";
};
```

Apoi trebuie să creăm acest fișier, pe care l-am declarat, ***/etc/bind/net.oretea.buna***, și să includem în el un conținut de acest gen (sintaxa trebuie respectată strict):

```
$TTL 1D

@ IN SOA oretea.buna. hostmaster.oretea.buna. (

202603121; serial

8H ; refresh
```

4H ; retry
4W ; expire
1D) ; minimum
IN A 10.1.1.8
;
@ IN NS euserveru.oretea.buna.

unlaptop IN A 10.1.1.125
euserveru IN A 10.1.1.8

Astfel, ne-am denumit calculatorul nostru ***euserveru***, și am mai dat nume unui laptop din rețea, ***unlaptop***, cu adresa 10.1.1.125. (Serialul ***202603121*** de mai sus e un număr oarecare, care trebuie modificat manual de fiecare dată când vrem să semnalizăm că am făcut o modificare în fișier)

După ce am salvat modificările în fișierele de configurare,
repornim serviciul ***bind9***:
sudo systemctl restart bind9
și îi verificăm starea:
systemctl status bind9

Dacă nu sunt semnalate erori, serviciul trebuie să funcționeze în rețea, și vom putea folosi denumirile de hosturi pe care le-am dat, ***euserveru.oretea.buna*** și, respectiv, ***unlaptop.oretea.buna*** (putem vedea, de ex., cu: **ping unlaptop.oretea.buna**)

sudo apt install samba – instalează serverul de fișiere
sudo systemctl stop smbd – oprește temporar daemon-ul, pentru configurarea lui.
Pentru a păstra fișierul de configurare original:
sudo mv /etc/samba/smb.conf /etc/samba/smb.conf.orig
Creăm un fișier de configurare nou:
sudo nano /etc/samba/smb.conf
cu următorul conținut:

[global]
server string = Server Fisiere
workgroup = WORKGROUP
security = user
map to guest = Bad user
name resolve order = host bcast wins
include = /etc/samba/smbshared.conf
Creăm apoi fișierul menționat, pentru continuarea configurării
sudo nano/etc/samba/smbshared.conf
[Documente]
path = /partajare/documente
force user = ferdi3
force group = grupferdi3
public = yes
writable = no

[Publice]
path = /partajare/publice
force user = ferdi3
force group = grupferdi3
create mask = 0664
force create mask = 0664
directory mask = 0777
force directory mask = 0777
public = yes
writable = yes

Creăm directorul partajare cu **sudo mkdir /partajare**, apoi cele două directoare în interiorul acestuia. În al doilea se va putea și scrie, în primul se va putea doar citi. Userul și grupul vor fi cele indicate: ferdi3 și grupferdi3 (dacă acestea există).

Cei din afară vor vedea la adresa //domeniulsauadresamea/Documente conținutul din /partajare/documente și la adresa //domeniulsauadresamea/Publice conținutul din /partajare/publice

Pentru aceasta, directorul /partajare trebuie deținut de ferdi3 și grupul grupferdi3:
sudo chown -R ferdi3:grupferdi3 /partajare

Pornim serverul de fișiere: **sudo systemctl start smbd**
Îi verificăm starea cu: **sudo systemctl status smbd**

Cele două directoare vor fi vizibile, la adresele respective în rețea și din Windows File Explorer (în bara de căutare putem tasta: \\domeniulsauadresamea\Documente)

sudo install apache2 – instalează
systemctl status apache2 – verifică dacă e pornit

fișierele site-ului se află în directorul /var/www/html (numit Document Root)

Certificatul de securitate Secure Sockets Layer (SSL)/Transport Layer Security (TSL)

Din oficiu, apache ascultă pentru trafic pe portul 80 (se poate vedea aceasta cu comanda **sudo ss -tulp | grep apache2**
– apare http, adică portul 80, și nu https, care folosește portul 443)

Pentru a activa suportul pentru traficul https, activăm modulul ssl:
sudo a2enmod ssl (cu **a2dismod** se dezactivează)
apoi, se repornește apache: **sudo systemctl restart apache2**

se editează fișierul de configurare
sudo nano /etc/apache2/sites-available/default-ssl.conf (care va trebui activat: **sudo a2ensite default-ssl.conf**)
Imediat sub prima linie (<VirtualHost _default_:443>) se adaugă opțiunea pentru ascultarea pe portul 443: **ServerName domeniulmeu.ro:443** (sau, dacă nu avem domeniu, adresa IP)

De asemenea, mai jos, în liniile referitoare
la SSLCertificateFile și SSLCertificateKeyFile,
trebuie trecute căile și fișierele reale unde se găsesc datele respective.

Varianta 1: certificat realizat de noi înșine

sudo mkdir /etc/apache2/certificat – creăm un director unde vom pune datele
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/certificat/domeniulmeu.ro.key -out /etc/apache2/certificat/domeniulmeu.ro.crt – generăm certificatul și cheia
Se dau răspunsurile cerute la prompt, apoi se va observa că s-au creat în directorul /etc/apache2/certificat două fișiere, domeniulmeu.ro.crt și domeniulmeu.ro.key.
Pe acestea le trecem în /etc/apache2/sites-available/default-ssl.conf: în locul celor două fișiere din oficiu, /etc/ssl/certs/ssl-cert-snakeoil.pem și

/etc/ssl/private/ssl-cert-snakeoil.key, vom avea deci
/etc/apache2/certificat/domeniulmeu.ro.crt și,
respectiv, /etc/apache2/certificat/domeniulmeu.ro.key.

sudo systemctl reload apache2 – se reîncarcă *apache*, cu noua configurație.

Traficul prin portul 443 va fi posibil, și va fi securizat. Totuși, browserele ar putea să trateze site-ul cu suspiciune, certificatul fiind creat de noi înșine.

Varianta 2: certificat obținut din exterior

O soluție simplă și gratuită o reprezită obținerea (și crearea) celor două fișiere cu ajutorul serviciului Let's Encrypt (letsencrypt.org/docs) se poate intra pe certbot.eff.org și se urmează pașii indicați (se instalează cu snap, așa cum se cere). Apoi, se trec fișierele nou-obținute în fișierul de configurare /etc/apache2/sites-available/default-ssl.conf

Utilizarea programului utilitar pentru firewall al Ubuntu Linux, *ufw*

Utilitarul se poate instala cu ***sudo apt install ufw***

Un firewall este un program care permite sau interzice accesul din exteriorul computerului la porturile de rețea. De obicei, firewall-urile interzic tot traficul cu exteriorul, din oficiu, în caz că nu sunt configurate altfel.

Starea programului se verifică cu una din comenzile:
systemctl status ufw sau ***sudo ufw status***
Programul poate activa cu:
sudo systemctl enable ufw sau ***sudo ufw enable***
și se poate dezactiva cu:
sudo systemctl disable ufw sau ***sudo ufw disable***
(activarea și dezactivarea se poate face și din consola *cockpit*)
Programul se poate porni sau reporni cu:
sudo systemctl start ufw
sudo systemctl restart ufw

Pentru stabilirea unor permisiuni de trafic, numite reguli (rules), utilitarul ***ufw*** acceptă comenzi având o sintaxă ca în exemplele următoare:
s***udo ufw allow from 192.168.1.156 to any port 22 (***permite trafic din partea adresei IP 192.168.1.156 prin portul 22)
sudo ufw allow 192.168.1.50 (permite trafic din partea adresei IP 192.168.1.50)
sudo ufw allow 80 (permite trafic prin portul 80)
sudo ufw allow 25/tcp (permite traficul prin portul 25, doar prin protocolul TCP - nu și prin UDP)
sudo allow ssh (permite traficul cu ssh prin portul folosit în mod obișnuit, 22)
sudo allow ftp (permite traficul cu ftp prin portul folosit în mod obișnuit, 21)
sudo allow http (permite traficul cu ftp prin portul folosit în mod obișnuit, 80)
sudo allow in http (***in*** e opțiunea subînțeleasă, în cazul ***http*** – e comanda minim necesară pentru vizitarea site-ului găzduit de noi; **out** e varianta în care

serverul nostru e cel care deschide site-uri cu *browser*-ul sau cu comenzi precum ***apt***).

sudo reject in http (respinge accesul prin http către calculatorul nostru)

ufw deny 80/tcp(interzice traficul prin portul 80, cu protocolul TCP)

ufw deny proto tcp to any port 80 (oprește tot traficul tcp prin portul 80)

ufw deny proto tcp from 192.168.10.0/24 to 192.168.0.1 port 25 (interzice tot traficul de la subrețeaua 192.168.10.0/24 la portul 25 al adresei IP 192.168.0.1)

Pentru a vedea regulile deja aflate în vigoare, sub o formă numerotată:

sudo ufw status numbered

Pentru a șterge una din reguli:

sudo ufw delete 2 (șterge regula cu numărul 2)

Ștergerea unei reguli se face și repetând modul în care a fost stabilită, adăugând înainte ***delete***:

sudo ufw delete allow 192.168.11.1

sudo ufw delete deny 80/tcp

În caz de contradicție între reguli, prioritate are regula cu numărul de ordine mai mic

Regulile ***ufw*** nu se pot modifica, ci, în caz de nevoie, ele se șterg, apoi se înlocuiesc cu altele noi.

Un manual în limba engleză pentru folosirea comenzii ***ufw*** se poate găsi la adresa:

https://manpages.ubuntu.com/manpages/noble/man8/ufw.8.html

Containerele reprezintă: un sistem de fișiere specific unui sistem de operare, dar fără kernel. Rulează izolat de restul sistemului, ca o mașină virtuală care nu are nevoie de resurse separate de cele ale sistemului de operare gazdă

Instalarea tehnologiei de containerizare Docker:
sudo apt install docker.io (instalează pachetul *docker.io*, din repository-ul oficial)
systemctl status docker (verificăm dacă rulează deja)
în caz că serviciul nu ne apare deja activat, îl putem activa cu:
systemctl enable --now docker

Ca să facem userul nostru curent (de ex. *ferdi11*) membru al grupului docker, grup secundar al său, dăm comanda:
sudo usermod –aG docker ferdi11
putem verifica rezultatul cu: ***groups ferdi11*** (afișează grupurile din care face parte *ferdi11*)

Verificăm că userul curent poate da acum într-adevăr comenzi cu *docker*:
docker version sau ***docker info***
(de obicei însă userii proaspăt adăugați trebuie să se relogheze, pentru a folosi noul grup)

Ca să instalăm o imagine de container a *Ubuntu* (oficială)
docker search ubuntu (căutăm în Docker Hub la https://hub.docker.com, între imagini gata create)
docker pull ubuntu (preluăm, sau *tragem* imaginea containerului Ubuntu, versiunea ultimă/*latest*)
docker images (vedem imaginile salvate pe PC, inclusiv imaginea *ubuntu*, cu ID-ul ei și cu tag-ul *latest*)
Pentru a șterge o imagine, putem folosi: ***docker rmi d3452s*** – argumentul comenzii fiind aici ID-ul prescurtat al imaginii respective)

Pentru a forma un container bazat pe imaginea instalată (ca dintr-un fel de șablon/*template*), cu ***docker run***:
docker run -it ubuntu /bin/bash
creează un container astfel încât să putem interacționa cu el ca și cu o mașină Ubuntu, prin shell-ul lui, în terminal; el va avea ca *entrypoint* (aplicația pe care un container e

configurat să o ruleze) shell-ul /bin/bash; din el se pot instala programe, dar când ieșim din *entrypoint* containerul dispare, cu tot cu programele instalate în el; comanda creează și un nume la întâmplare pentru container; dacă vrem să i-l dăm noi, inserăm opțiunea --name *unnume*:

docker run -it --unnume ubuntu /bin/bash

putem porni containerul fie cu
docker start idcontainerpescurt
sau cu ***docker start numecontaintercomplet***
(cu ***stop*** în loc de ***start*** îl oprim)

Pentru a ne deconecta de la el (și astfel oprindu-l), folosim:
exit sau apăsăm ***Ctrl+d***

Pentru a ne detașa din el (ieșind din shell-ul lui fără să îl oprim), apăsăm
Ctrl+p+q

Putem vedea containerele care rulează cu:
docker ps
Putem vedea toate containerele, fie rulate, fie oprite (dar care pot fi repornite) cu:
docker ps -a

Pentru a crea un container care rulează în fundal, ca un serviciu, adăugăm opțiunea ***d*:**
docker run -dit ubuntu /bin/bash
d înseamnă în modul „detașat” adică îl va rula în fundal, și nu se va opri până nu i se va cere)
i în modul interactiv
t cu un terminal (pseudo tty)
ubuntu e numele imaginii după care se creează containerul
/bin/bash înseamnă că el va rula shell-ul bash

Crearea unui container cu redirecționarea portului 80 al său către portul 8080 al gazdei, pentru a putea accesa containterul cu un browser
docker run -dit -p 8080:80 ubuntu /bin/bash
în urma comenzii, se va arăta ID-ul (întreg) al containerului
Pentru a ne atașa la el, odată pornit vom da aceeași comandă.
docker attach IDcontainer
acum, că am intrat în container, îi putem instala de pildă serverul web apache:
apt update
apt install apache2 (suntem deja root în container, deci nu e nevoie și de comanda sudo)
/etc/init.d/apache2 start (așa se pornește apache2 în containter, nu cu systemctl start apache2, pentru că systemctl nu va funcționa; pentru fiecare aplicație, trebuie citită documentația despre cum se pornește în container)

Putem transforma containerul într-o nouă imagine cu comanda:

docker commit idcontainer numeimnou:tag

putem da un tag precum *1.0*, sau dacă scriem doar numele, subînțelege că tag-ul va fi *latest*.

În loc de id-ul containerului (e de ajuns începutul lui) putem trece numele lui (dar complet)

Putem vedea că s-a creat o nouă imagine, cu: ***docker images***

În el putem constata, cu ajutorul comenzii ***ps ax*** că rulează puține procese (doar această comandă, plus bash și apache)

Persistența datelor după închiderea containerului

Putem lega conținutul unui director al gazdei (sursă) cu cel al unui director din container (țintă): ținta se șterge și va prelua sursa, iar lafinal modificările făcute de container vor rămâne salvate în directorul sursă.

docker run -dit -p 8081:80 --mount type=bind,source=/var/www/html,target=/var/www/html numeimnou:tag /bin/bash

Salvarea și mutarea unei imagini

Putem salva (sub formă de arhivă) o imagine, pentru a o muta eventual pe alt calculator

docker save -o numeimptmutat.tar numeimnou:tag

(îl va salva în directorul curent, dar oricum l-am găsi cu: ***sudo find / numeimptmutat* | grep numeimptmutat***)

cu ***scp*** sau ***scp -r*** putem copia cu ssh fișierul sau și directorul cu date pe alt calculator

apoi, pentru putea prelua imaginea pe al doilea calculator, folosim:

docker load –i /calea/unde/l-am/copiat/numeimptmutat.tar

Virtual Host cu Apache (două adrese de site pe aceeași adresă IP)

Fișierul de configurare al Apache este împărțit de Ubuntu în mai multe secțiuni: o parte principală, de la care se face trimitere și la alte fișiere secundare

Îl putem găsi afișând fișierele instalate pentru apache cu

dpkg –L apache2

între ele îl remarcăm pe /etc/apache2/apache2.conf și îl deschidem cu

sudo nano /etc/apache2/apache2.conf

Userul sub care rulează procesul apache2 este www-data.

Acest user trebuie să aibă permisiuni pentru fișierele site-ului

E un pas care trebuie făcut la instalarea unui site: după ce copiezi fișirele wordpress sau joomla în Documentroot trebuie să le schimbi ownership-ul la identitatea sub care rulezi apache-ul: ***chown -R www-data:www-data /var/www/***

Include configurări pentru module, pentru porturile pe care ascultă

La sfârșitul fișierului observăm că sunt incluse fișierele *conf-enabled/*.conf* și *sites-enabled/*.conf*

În directorul *sites-enabled* putem avea mai multe fișiere de configurare, nu doar cel din oficiu, 000-default.conf

În acesta vedem cu ***sudo nano 000-default.conf***

că sunt definiți, după linia *<VirtualHost *:80>*

parametrii *ServerName* (numele sau adresa sub care se deschide site-ul)

și DocumentRoot (directorul unde se află fișierele site-ului)

De exemplu, aceste prime rânduri ar fi, într-un fișier de configurare site1.conf

*<VirtualHost *:80>*

ServerName www.siteulmeu1.ro

DocumentRoot /var/www/html1

Putem dezactiva sau nu fișierul din oficiu cu

sudo a2dissite 000-default.conf

și le activăm pe cele pe care le creăm noi, după același model:

sudo a2ensite site1.conf și, eventual, ***sudo a2ensite site2.conf***

După ce repormim apache2, putem apela site-ul din directorul /var/www/html1 cu adresa www.siteulmeu1.ro și vom vedea varianta de site nouă. Pentru a tasta în browserul local numele de domeniu nou, îl trecem, alături de adresa IP în fișerul /etc/hosts. Apoi putem verifica în browser dacă site-ul se deschide. La fel, putem face și cu un al doilea site, www. siteulmeu2.ro, pentru care fișierele ar fi în */var/www/html2*.

sudo apt update
sudo apt install mariadb-server (instalăm MariaDB)
systemctl status mariadb (verificăm că este pornit – în container Ubuntu, **service mariadb status**)

Adăugăm elemente de securitate serverului (dacă nu suntem în container):
sudo mysql_secure_installation
Aici avem de răspuns la întrebări, la prompt (am dat răspunsurile recomandate, pentru început)
Switch to unix_socket authentication [Y/n] ***n***
Change the root password? [Y/n] ***y***
Trebuie să dăm o parolă de root pentru MariaDB (nu root pentru tot sistemul)
și apoi să confirmăm parola
de ex: ***parola123***
parola123
Remove anonymous users? [Y/n] ***y***
Disallow root login remotely? [Y/n] ***y***
Remove test database and access to it? [Y/n] ***y***
Reload privilege tables now? [Y/n] ***y***
Acum, că serverul a fost instalat complet, ca să intrăm în shell-ul MariaDB, scriem:
sudo mariadb sau
mariadb -u root -p (în container Ubuntu, folosim **service mariadb start**)
După ce ni se cere parola de root, primim acces în shell, și vom vedea promptul
MariaDB [(none)]>
Putem deci crea sau șterge baze dedate, useri, tabele. E convenabil însă ca acum să creăm un user pentru a gestiona serverul MariaDB, ca să nu lucrăm direct cu root-ul
ieșirea din shell se face cu ***exit*** și Enter sau cu ***Ctrl+d***
Fișierele de configurare ale MariaDB se găsesc în /etc/mysql
În shell-ul MariaDB putem crea un user
MariaDB [(none)]> ***CREATE USER 'userdewp'@'%' IDENTIFIED BY 'Parola-de-WP';***
FLUSH PRIVILEGES;
(Se scriu comenzile cu majuscule ca să fie mai ușor de deosebit de date, dar nu e obligatoriu.)

În loc de '%' putem scrie 'localhost', dacă vrem ca userul *userdewp* să se poată loga nu de *oriunde*, ci doar *local*, sau putem scrie '192.168.1.%' dacă vrem să se poată face logare doar *din această rețea*.

Comanda FLUSH PRIVILEGES trebuie dată *întotdeauna* când se adaugă un user sau se schimbă permisiuni,

Atribuim acum toate privilegiile către userul creat, când se loghează de oriunde sau de la nivel local:

GRANT ALL PRIVILEGES ON *.* TO 'userdewp'@'%'; sau ***@'localhost'***

FLUSH PRIVILEGES;

Acum se poate intra în shell și cu userul nou creat:

exit

mariadb -u userdewp -p și introducem parola aleasă, ***Parola-de-WP***

sau ne putem loga direct cu ***mariadb -u userdewp -pParola-de-WP***

(dar ni se va putea vedea parola în istoricul comenzilor Linux)

Se poate crea și un user care să poată doar citi date, fără a face modificări. Pentru a face aceasta dintr-o dată, dăm comanda următoare, în shell-ul MariaDB, ca root:

GRANT SELECT ON *.* TO 'userdecitit'@'localhost' IDENTIFIED BY 'Parolacitit';

FLUSH PRIVILEGES;

Pentru a crea o bază de date, dăm, ca root, comanda:

CREATE DATABASE bazadatewp;

O putem vedea apoi cu:

SHOW DATABASES;

Pentru a vedea și userii:

SELECT HOST, USER, PASSWORD FROM mysql.user;

De obicei se creează întâi baza de date, și apoi userul care are acces doar la acea bază de date.

GRANT SELECT ON bazadatewp.* TO 'usercititor'@'localhost' IDENTIFIED BY 'parolacititor';

sau ***GRANT ALL ON bazadatewp.* TO 'useradmin'@'localhost' IDENTIFIED BY 'parolaadmin';***

verificăm ce am făcut cu

SHOW GRANTS FOR 'useradmin'@'localhost';

În loc de ALL, putem trece permisiunile CREATE, DELETE, INSERT, SELECT sau DROP, pentru adăugat rânduri în baza de date, pentru șters rânduri din baza de date, pentru adăugat rânduri într-un tabel, pentru citit informații din baza de date, respectiv pentru șters cu totul o bază de date. (Există de altfel și alte permisiuni, mai avansate.)

Pentru șters un user, folosim:

DROP USER 'usercititor'@'localhost'

Pentru a crea un tabel în baza de date, ne logăm cu un user care are toate privilegiile cu referire la baza de date respectivă, apoi scriem:

USE bazadatewp;

CREATE TABLE Angajati (Nume char(15), Varsta int(3), Pozitie char(15));

se poate verifica rezultatul cu:
SHOW COLUMNS IN Angajati;
Am creat deci un tabel cu numele Angajati, cu trei coloane: Nume, Varsta si Pozitie
Putem adăuga date în tabel cu comanda INSERT:
INSERT INTO Angajati VALUES ('Ionut Popovici', '32', 'Paznic');

Se pot citi toate datele din tabel cu:
SELECT * FROM Angajati;
Se poate șterge o intrare din tabel cu:
DELETE FROM Angajati WHERE Nume = 'Ionut Popovici';
Se poate sterge tabelul cu:
DROP TABLE Angajati;
Sau se poate sterge baza de date cu:
DROP DATABASE bazadatewp;

Pentru salvare (backup) pentru o bază de date (comandă dată din shell-ul Linux):
mysqldump -u userdewp -p --database bazadatewp > bazadatewp.sql

Refacerea bazei de date din acel backup se realizează cu:
sudo mariadb -u userdewp -p < bazadatewp.sql

Ieșirea din shell-ul MariaDB se face cu **Ctrl+d**

(la fel se procedează și pentru WordPress)

Avem de instalat întâi modulul php împreună cu extensiile necesare

sudo apt install php php-xml php-mysql php-mbstring php-mbstring php-zip php-soap php-sqlite3 php-curl php-gd php-ldap php-common

(verificăm versiunea instalată cu ***php -v***)

Descărcăm kitul joomla:

sudo wget https://downloads.joomla.org/cms/joomla6/6-0-3/Joomla_6-0-3-Stable-Full_Package.zip?format=zip

Apoi,pentru a dezarhiva,

sudo apt install unzip

sau, dacă funcționează deja comanda unzip,

dăm direct ***unzip*** la fisierul descărcat în directorul /var/www/joomla

Apoi intrăm în directorul respectiv cu ***cd***, ca să schimbăm permisiunile de fișier pentru tot directorul, cu opțiunea -R (adică recursiv): vom pune ca deținător userul www-data, sub care rulează procesul Apache

sudo chown -R www-data: ./

schimbăm și permisiunile astfel:

sudo chmod -R 755 ./

Va trebui să creăm și un virtual host al Apache pentru joomla, cu

sudo nano /etc/apache2/sites-available/joomla.conf

Și avem scris în acest fișier de configurare:

<VirtualHost *:80>
ServerName ferdinand.go.ro
DocumentRoot /var/www/joomla/
</VirtualHost>

Apoi dezactivăm 000-default.conf

a2dissite 000-default.conf

Și activăm joomla.conf

sudo a2ensite joomla.conf

Și activăm modulul rewrite

sudo a2enmod rewrite

În fine, putem reporni Apache:

sudo systemctl restart apache2

Accesăm Joomla Web Installer, deschizând adresa setată, ferdinand.go.ro, sau IP-ul nostru într-un web browser, și setăm userul și parola pentru baza noastră de date instalată în prealabil. Apoi dăm nume de user și parolă pentru user admin.Și putem folosi Joomla pentru crearea site-ului, în back-end, cu ajutorul unui tutorial.

Pentru pornirea Pentru persistența datelor pe calculatorul gazdă, creăm directoare în care să salvăm datele din contaier, și pe care același container sau altul asemănător să le încarce la pornire.

Anterior se copiază din container conținutul directoarelor, pe un host, cu comenzi precum:

docker cp 268:/etc/apache2/ etcapache2

(în loc de 268 se trece id-ul respectivul container activ, ia etcapache2 e o variantă de nume de folder ușor memorabilă în care să se păstreze datele din acea cale din container).

Datele de pe directoarele hostului se transferă pe alt calculator tot de la linia de comandă, cu ***scp***, iar nu în modul grafic.

Pentru copiere trebuie să avem permisiunile necesare pentru fișierelor. De asemenea, la pornirea noului container, trebuie să avem setat pentru **/var/lib/mariadb**, deci pentru directorul corespunzător de pe host (de pildă, **varlibmariadb**), ca userul deținător să fie *mysql*: se va căuta id-ul acestui user în containerul sursă cu (***cat /etc/passwd | grep mysql***), și se va seta același *id* de user și de grup (de pildă, putem găsi că este 994) pentru pentru folderul din host (cu ***sudo chown -R 994:994 varlibmariadb***)

Containerul deci se pornește cu o comandă completă precum:

docker run -dit -p 80:80 --name joompersist -v /home/ferdi5/etcapache2:/etc/apache2 -v /home/ferdi5/varwwwhtml:/var/www/html -v /home/ferdi5/varwwwjoomla:/var/www/joomla -v /home/ferdi5/varlogapache2:/var/log/apache2 -v /home/ferdi5/varlibmariadb:/var/lib/mariadb ubuapamarjoo364:1.0 /bin/bash

(presupunând că *ubuapamarjoo364:1.0* este numele imaginii de container, *joompersist* este numele pe care îl dăm noului container, iar directoarele pentru persistență le avem în directorul home al userului *ferdi5*)

După pornirea containerului, pornim în el apache și mariadb (cu ***service apache2 start*** și ***service mariadb start***), iar site-ul trebuie să fie funcțional, pe adresa hostului.

Cum ne conectăm prin *ssh* cu cheie la o instanță EC2, în AWS

Cheia pentru conectarea la o instanță Ubuntu Server am numit-o *chee*. Ea se descarcă automat,cu extensia *.pem*, în directorul ~/Downloads.

Intrăm în director și mai întâi îi restrângem permsiunile astfel:

sudo chmod 400 chee.pem

Verificăm rezultatul cu:

ls -l chee.pem

Pentru a ne conecta prin *ssh* (numele de user fiind *ubuntu*), folosim apoi una din formele de mai jos de comandă,folosind adresa IP publică sau numele de domeniu primit pentru instanța noastră:

ssh -i chee.pem ubuntu@16.171.54.24

sau:

ssh -i "chee.pem" ubuntu@ec2-16-171-54-24.eu-north-1.compute.amazonaws.com

Crearea unui domeniu funcțional pentru o rețea locală cu Windows Server

A. Instalarea AD DS, DNS, DHCP pe server

1. Redenumirea calculatorului[1], pentru adresarea mai ușoară.

Opțiunea de redenumire se accesează pe mai multe căi. De ex: **Butonul de Start (cu pictograma Windows)>Settings>About>Rename PC** sau **Bara de căutare>View PC Name> Rename PC** (apoi, restartare)

2. Conectarea fizică la rețea, prin patch-cord-uri și switch alimentat la priză.

3. Alocarea unei adrese IP statice:

Click pe pictograma de rețea (în bara de stare)>(Network and Sharing Center)> Change Adapter Settings>Ethernet>Internet Protocol Version4 (dublu click).

Aceeași adresă IP se alocă și pentru serverul DNS

(Pentru eventuala cuplare la internet, identificatorul rețelei din adresa router-ului trebuie păstrat și pentru adresa serverului nostru; la gateway se trece adresa routerului, iar la a doua adresă DNS se trece serverul DNS al furnizorului nostru de internet, sau o adresă publică DNS cunoscută, de ex. 8.8.8.8, de la Google)

Închidem ferestrele (**OK**, **OK**)

Verificăm, cu **Bara de căutare>cmd>tastăm la prompter: ipconfig/all**

4. Instalarea rolurilor Active Directory Domain Services, Domain Name System (DNS) Server, și Dynamic Host Configuration Protocol (DHCP) Server:

(Butonul Start>)Click pe pictograma Server Manager>Manage>Add Roles and Features>Next>Next>Next (serverul nostru e deja selectat)**>Click pe Active Directory Domain Services>În** fereastra care apare, **Click pe Add Features>Click și pe DNS și DHCP,** și verificăm ca Group Policy Management să fie și selectat**)>Next>Next>Install**

(Atenție, nu se închide fereastra în timpul instalării, sau după instalare!)

Click pe Promote this server to a domain controller (în Windows Server 2008 dăm aici comanda **DCPromo)>Add a new forest** și scriem numele domeniului **(de forma: xxxx.xxxx)>Next>**Selectăm un nivel de funcționalități pentru forest și pentru domeniu (dacă vom lucra și cu DC mai vechi în rețea putem opta pt minim) și alegem o parolă pentru domeniu**>Next>Next>Next>Next>Next**>(dacă totul e corect, apare butonul:) **Install>Close** (se va restarta computerul)

[1] De la Windows Server 2012, redenumirea computerului și accesul la setările Ethernet se găsesc și în **(Butonul Start>)(pictograma)Server Manager>Local Server**

La repornire, introducem parola aleasă.

5. Configurări

Verificăm că avem instalate rolurile în: **Start>Administrative Tools** și le alegem pe rând, pentru configurare

În **Active Directory Users and Computers**, creăm o Organizational Unit, și în interior un user

În **DHCP**, creăm un Scope – verificăm că e activat, autorizat. Verificăm la **IPv4>Server Options>Configure Options** setările pentru DNS Server (006) și DNS Domain Name (015)

În **DNS** verificăm traducerea corectă a adresei IP a Name Server-ului

La fiecare schimbare de setare, se va excuta și (click pe butonul) Refresh

B. Conectarea unui PC client la domeniu

1. Reținem numele domeniului, și adresa IP a serverului - se pot vedea și cu cmd>ipconfig/all, și (din 2012) în **Server Manager >Local Server**

2. *La PC-ul client, introducem adresa serverului DNS* (al domeniului, și, alternativ, al unui server DNS public sau al provider-ului de internet)în meniul **(Change Adapter Settings>Ethernet>)Internet Protocol Version4**. Închidem ferestrele, cu **OK**.

3. *Verificăm conectivitatea cu serverul*, prin **cmd>ping** (nume server sau adresa IP server), după ce ne-am asigurat că am făcut calculatoarele vizibile pentru rețea (de ex. în **File Manager**).

4. *Se schimbă numele PC-ului*, în **Settings>About>Rename this PC (Advanced)>Computer Name Change>Member of…** introducându-se numele domeniului.

La introducerea numelui și parolei de user, PC-ul va fi acceptat și se va restarta pentru conectarea la domeniu.

(5. Ieșirea din domeniu se va face doar cu schimbarea adresei serverului DNS – permisă de administratorul serverului actual. Logarea la alt domeniu sau la calculatorul neconectat la nici un domeniu se face adăugând ca prefix la userul respectiv fie numele domeniului nou, fie al calculatorului: de ex: domeniunou\usernou, sau PC-01\Ferdi8)